WebFaction API Documentation

Swarma Limited - WebFaction is a service of Swarma Limited

CONTENTS

1 Introduction **3**

2 Tutorial **5**
- 2.1 Getting Started . . . 5
- 2.2 Creating an Email Address . . . 5
- 2.3 Installing an Application . . . 6
- 2.4 Packaging the Install Script for the Control Panel . . . 7
- 2.5 Additional Resources . . . 8

3 API Reference **9**
- 3.1 General . . . 9
- 3.2 Email . . . 10
- 3.3 Websites and Domains . . . 14
- 3.4 Applications . . . 18
- 3.5 Cron . . . 19
- 3.6 DNS . . . 20
- 3.7 Databases . . . 21
- 3.8 Files . . . 24
- 3.9 Shell Users . . . 25
- 3.10 Servers . . . 26
- 3.11 Miscellaneous . . . 26

4 Application Types **29**

Index **45**

Contents:

CHAPTER

ONE

INTRODUCTION

The WebFaction API (Application Programming Interface) is a powerful XML-RPC interface for managing many control panel and account tasks. With the WebFaction API, you can automate application installation, email address configuration, and more.

Like other XML-RPC APIs, the WebFaction API works by sending a short piece of XML over HTTP. Luckily, many languages have XML-RPC libraries to make requests quick and painless.

For example, you can send an XML-RPC request using Python's xmlrpclib module:

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('widgetsco', 'widgetsrock')
```

Or with Ruby's `xmlrpc` package:

```
>> require 'xmlrpc/client'
=> true
>> require 'pp'
=> true
>> server = XMLRPC::Client.new2("https://api.webfaction.com/")
#<XMLRPC::Client:0x5b1698 @cookie=nil, @create=nil, @port=443>
>> pp server.call("login", "widgetsco", "widgetsrock")
["ca4c008c24c0de9c9c8",
 {"mail_server"=>"Mail5",
  "web_server"=>"Web55",
  "username"=>"widgetsco",
  "id"=>687,
  "home"=>"/home"}]
=> nil
```

To learn more about XML-RPC and find an implementation in your favorite language, please visit XMLRPC.com.

CHAPTER

TWO

TUTORIAL

The WebFaction API allows you to write scripts to automate certain tasks that you would normally accomplish with the control panel or an SSH session.

For instance, you could use the API to write a script to configure lots of email addresses, instead of creating them one by one in the control panel.

You can also use the API to automate the installation of any application that you like, and you can share the resulting install script to allow other users to use it in one click.

2.1 Getting Started

The API is a set of methods available via XML-RPC calls at the URL `https://api.webfaction.com/`. In this documentation we will use the Python programming language to talk to the API, but you can use any language that you want, provided that it has an XML-RPC library.

First, connect to the server and login:

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('test5', 'password')
>>> account
{'username': 'test5', 'home': '/home2', 'id': 237}
```

The username and password passed to the login method are those used to login to the control panel.

As you can see, the `login` method returns a tuple. The first element is a string containing a session ID that you will need to pass to all subsequent methods. The second element is a dictionary containing various data about your account, including the base home directory (usually `/home` or `/home2`).

2.2 Creating an Email Address

Let's create a new email address for our account:

```
>>> server.create_email(session_id, 'user@mydomain.example', 'test5')
{'email_address': 'user@mydomain.example', 'id': 2037, 'targets': 'test5'}
```

The `create_email` method takes the session ID, the email address and a string of comma separated target mailboxes. It returns a dictionary containing various data about the newly created email address.

This call is equivalent to creating the email address from the control panel, but the advantage is that you can script it.

2.3 Installing an Application

Now let's see which methods we can use to install an application. For this tutorial, let's install a Joomla application.

First, create a `Static/CGI/PHP` application:

```
>>> server.create_app(session_id, 'my_joomla_app', 'static', False, '', False)
{'app_type': 'static',
 'name': 'my_joomla_app',
 'id': 3901,
 'autostart': False,
 'port': 0,
 'open_port': False,
 'extra_info': ''}
```

`create_app` uses the following parameters:

- *session_id* – session ID returned by `login`
- *name* (string) – name of the application
- *type* (string) – type of the application
- *autostart* (boolean) – whether the app should get restarted with an `autostart.cgi` script
- *extra_info* (string) – additional information required by the application (for example, a file path). If `extra_info` is not required by the application, it is ignored.
- *open_port* (boolean) – for applications that listen on a port, whether the port should be open on shared and dedicated IP addresses

Calling `create_app` is equivalent to creating the application through the control panel: it creates the directory and configures everything that's needed for this application.

Next, you need to download the Joomla archive and extract it in the app directory. To do so we'll use the `system` method of the API, which allows us to execute some command on our server as if we were logged in with SSH. Since the default Static/CGI/PHP app comes with an `index.html` file which would shadow the `index.php` file provided by Joomla, we will also delete that file:

```
>>> cmd = "rm -f index.html;"
>>> cmd += "wget https://wiki.webfaction.com/attachment/wiki/JoomlaFiles/Joomla_1.5.0-Beta-Full_Packa
>>> cmd += "tar xzvf Joomla_1.5.0-Beta-Full_Package.tar.gz?format=raw"
>>> server.system(session_id,  cmd)
''
```

Because we previously installed an application, the `system` method automatically runs in the new application's directory. The `system` method returns whatever the command printed to standard output. If the command prints something to standard error, `system` raises an error with that text.

Next, create a MySQL database, since Joomla requires one:

```
>>> server.create_db(session_id, 'test5_joomla_db', 'mysql', 'db_password')
{'type': 'mysql',
'id': '116101115116053095106111111109108097095100098045l',
'name':'test5_joomla_db'}
```

This creates a MySQL database called `test5_joomla_db` and a user of the same name, with the password `db_password`.

Next, copy the Joomla configuration file from `configuration.php-dist` to `configuration.php` and then edit it to specify the database connection settings:

```
>>> server.system(session_id, "cp configuration.php-dist configuration.php;")
''
>>> server.replace_in_file(session_id, 'configuration.php',
        ("var $user = '';", "var $user = 'test5_joomla_db';"),
        ("var $password = '';", "var $password = 'db_password';"),
        ("var $db = '';", "var $db = 'test5_joomla_db';"))
''
```

The handy `replace_in_file` method lets us find and replace in a file. It takes a session ID, the name of the file, and any number of tuples containing a string to replace and the replacement string.

There are a few other steps needed to install a Joomla application: we need to edit the Joomla SQL file and run it. We won't detail them here—you can look at the actual script for details.

2.4 Packaging the Install Script for the Control Panel

Previously, we installed a Joomla application by manually running a bunch of commands. We can package these commands in an install script for the control panel to run for us. The advantage is that we can then run this install script over and over again directly from the control panel or share it with others.

2.4.1 How the Install Script is Run

To use an install script in the control panel:

1. Log in to the control panel.
2. Click *Domains / websites* → *Applications*. The list of applications appears.
3. Click the *Add new application* button. The *Create a new application* form appears.
4. In the *Name* field, enter a name for the application.
5. In the *App Category* menu, click to select *Custom*. The *Script URL* field appears.
6. In the *Script URL* field, enter the install script's URL.
7. Click the *Fetch URL* button.
8. If applicable, in the *Machine* menu, select a web server.
9. Click the *Save* button.

When you click the *Create* button, the script is run with the following parameters:

install_script `create|delete` *username password app_name autostart extra_info*

- *username* – control panel username
- *password* – user's hashed password
- *app_name* – application name from *Name* field.
- *autostart* (boolean) – whether the user selected the *Autostart* checkbox
- *extra_info* – contents of the *Extra info* field.

When the user creates the app, the control panel will call the script with `create` as the first parameter. If the user deletes the app later on, the control panel will call the script with `delete` as the first parameter.

When it calls the script with `create`, the control panel expects the script to print the application ID to standard output and nothing else. If the script prints anything else to standard output, or prints anything to standard error, the control panel will display it as an error message on the add application page.

When it calls the script with `delete`, the control panel expects your script to not print anything to standard output or standard error. If anything gets printed the control panel will display it as an error message.

Additionally, if your script is written in Python and you include a docstring (**PEP 257**) at the beginning of the script, the control panel will use it as the application documentation.

To see two examples of install scripts, take a look a the Joomla install script or the MoinMoin install script.

2.4.2 Making a Script Available in One Click

Now that you have your install script, one way to share it would be to ask others copy and paste it in the control panel. That's not convenient but fortunately, there is a better way.

On the add application page in the control panel, there is a field *Install script url* that appears when you select *Custom install script*. If you enter a URL and click *Fetch URL*, the control panel will look for an install script at that URL. For the control panel to find the script on the page, your script must be enclosed between these two magic tags:

```
-----BEGIN WEBFACTION INSTALL SCRIPT-----
-----END WEBFACTION INSTALL SCRIPT-----
```

If you make your script available on the web, all you have to do to let anyone use it is give them a special URL in the form `https://my.webfaction.com/app/new-application?script_url=location` where *location* is the escaped URL to your script.

You can see examples at InstallScripts, where Joomla, MoinMoin, and other applications can be installed using that approach.

2.5 Additional Resources

For more information about the WebFaction API, please consult the *API Reference*.

CHAPTER

THREE

API REFERENCE

The WebFaction XML-RPC API provides methods to handle many account tasks. This documentation is a complete reference to all of the possible API methods.

Please note that XML-RPC parameters are positional (order matters), and many parameters are required. Parameters may only be omitted if omitted parameters have default values and follow all other parameters to which you have supplied a value.

3.1 General

`login`

Parameters

- **username** (*string*) – a valid WebFaction control panel username
- **password** (*string*) – a valid WebFaction control panel user's password
- **machine** (*string*) – the case-sensitive machine name (for example, `Web55`); optional for accounts with only one machine

Log in a user and return credentials required to make requests for that user. The method returns a session ID string and a struct containing following key-value pairs:

id account ID

username username

home home directory path

web_server Web server associated with the logged in account (for example, `Web55`)

mail_server mail server associated with the logged in account (for example, `Mailbox2`)

Note: The session ID is required for all subsequent API calls.

`list_disk_usage`

Parameters **session_id** – session ID returned by `login`

List disk space usage statistics about your account (similar to usage statistics shown on the control panel). The method returns a struct containing the following members:

`home_directories` A list of structs with details for each home directory associated with the account. Each struct contains the following members:

`last_reading` The date and time of the last recording of the home directory's size

`machine` The server name (for example, `Web300`)

`name` The username

`size` The disk usage in kilobytes

`mailboxes` A list of structs with details for each mailbox associated with the account. Each struct contains the following members:

`last_reading` The date and time of the last recording of the mailbox's size

`name` The mailbox name

`size` The disk usage in kilobytes

`mysql_databases` A list of structs with details for each MySQL database associated with the account. Each struct contains the following members:

`last_reading` The date and time of the last recording of the database's size

`name` The database name

`size` The disk usage in kilobytes

`postgresql_databases` A list of structs with details for each PostgreSQL database associated with the account. Each struct contains the following members:

`last_reading` The date and time of the last recording of the database's size

`name` The database name

`size` The disk usage in kilobytes

`total` The account's total disk usage in kilobytes

`quota` The account's total disk allotment in kilobytes

`percentage` The account's total disk usage as a percentage of the quota (for example, an account using 3.1 GB of 100 GB would use `3.1` percent of its quota)

3.2 Email

3.2.1 Mailboxes

`change_mailbox_password`

Parameters

- **session_id** – session ID returned by `login`
- **mailbox** (*string*) – a valid mailbox name
- **password** (*string*) – the new mailbox password

Change a mailbox password.

See Also:

See Strengthening Passwords for important information about choosing passwords.

`create_mailbox`

Parameters

- **session_id** – session ID returned by `login`
- **mailbox** (*string*) – mailbox name
- **enable_spam_protection** (*boolean*) – whether spam protection is enabled for the mailbox (optional, default: true)
- **discard_spam** (*boolean*) – whether spam messages received by the new mailbox are discarded (optional, default: false)
- **spam_redirect_folder** (*string*) – name of the IMAP folder where messages identified as spam are stored (optional, default: an empty string)
- **use_manual_procmailrc** (*boolean*) – whether to use manual procmailrc rules as specified by the `manual_procmailrc` parameter (optional, default: false)
- **manual_procmailrc** (*string*) – the procmailrc rules for the mailbox (optional, default: an empty string)

> **Warning:** If `discard_spam` is true, messages misidentified as spam—false positives—may be lost permanently.

Create a mailbox and return a struct containing the following key-value pairs:

id mailbox ID

name mailbox name

enable_spam_protection name of the folder where messages identified as spam are stored

password a randomly generated password

discard_spam a boolean indicating whether spam emails are be discarded

spam_redirect_folder name of the IMAP folder where messages identified as spam are stored

use_manual_procmailrc a boolean indicating whether manual procmailrc rules are enabled

manual_procmailrc a string containing manual procmailrc rules

See Also:

`update_mailbox`

`delete_mailbox`

Parameters

- **session_id** – session ID returned by `login`
- **mailbox** (*string*) – mailbox name

Delete a mailbox.

`list_mailboxes`

Parameters **session_id** – session ID returned by `login`

Get information about the account's mailboxes. The method returns an array of structs with the following key-value pairs:

id mailbox ID

name mailbox name

enable_spam_protection name of the folder where messages identified as spam are stored

password a randomly generated password

discard_spam a boolean indicating whether spam emails are be discarded

spam_redirect_folder name of the IMAP folder where messages identified as spam are stored

use_manual_procmailrc a boolean indicating whether manual procmailrc rules are enabled

manual_procmailrc a string containing manual procmailrc rules

`update_mailbox`

Parameters

- **session_id** – session ID returned by `login`
- **mailbox** (*string*) – mailbox name
- **enable_spam_protection** (*boolean*) – whether spam protection is enabled for the mailbox (optional, default: true)
- **discard_spam** (*boolean*) – whether spam messages received by the new mailbox are discarded (optional, default: false)
- **spam_redirect_folder** (*string*) – name of the IMAP folder where messages identified as spam are stored (optional, default: an empty string)
- **use_manual_procmailrc** (*boolean*) – whether to use manual procmailrc rules as specified by the `manual_procmailrc` parameter (optional, default: false)
- **manual_procmailrc** (*string*) – the procmailrc rules for the mailbox (optional, default: an empty string)

> **Warning:** If `discard_spam` is true, messages misidentified as spam—false positives—may be lost permanently.

Change the details of an existing mailbox. The mailbox must exist before calling the method. The method returns a struct containing the following key-value pairs:

id mailbox ID

name mailbox name

enable_spam_protection name of the folder where messages identified as spam are stored

password a randomly generated password

discard_spam a boolean indicating whether spam emails are be discarded

spam_redirect_folder name of the IMAP folder where messages identified as spam are stored

use_manual_procmailrc a boolean indicating whether manual procmailrc rules are enabled

manual_procmailrc a string containing manual procmailrc rules

See Also:

`create_mailbox`

3.2.2 Addresses

`create_email`

Parameters

- **session_id** – session ID returned by `login`
- **email_address** (*string*) – an email address (for example, `name@example.com`)
- **targets** (*string*) – names of destination mailboxes or addresses, separated by commas
- **autoresponder_on** (*boolean*) – whether an autoresponder is enabled for the address (optional, default: false)
- **autoresponder_subject** (*string*) – subject line of the autoresponder message (optional, default: an empty string)
- **autoresponder_message** (*string*) – body of the autoresponder message (optional, default: an empty string)
- **autoresponder_from** (*string*) – originating address of the autoresponder message (optional, default: an empty string)
- **script_machine** (*string*) – a machine name for specifying a path to a script (optional, default: an empty string)
- **script_path** (*string*) – an absolute path to a script; see *Sending Mail to a Script* for details (optional, default: an empty string)

Create an email address which delivers to the specified mailboxes.

If *autoresponder_on* is true, then an autoresponder subject, message, and from address may be specified.

See Also:

`update_email`

`delete_email`

Parameters

- **session_id** – session ID returned by `login`
- **email_address** (*string*) – an email address (for example, `name@example.com`)

Delete an email address.

`list_emails`

Parameters **session_id** – session ID returned by `login`

Get information about the account's email addresses. The method returns an array of structs with the following key-value pairs:

id email ID

email_address email address

targets mailboxes or email addresses to which the address is set to deliver

autoresponder_on a boolean indicating whether an autoresponder is enabled for the address

autoresponder_subject the autoresponder subject line (if applicable)

autoresponder_message the autoresponder message body (if applicable)

autoresponder_from the autoresponder from address (if applicable)

`update_email`

Parameters

- **session_id** – session ID returned by `login`
- **email_address** (*string*) – an email address (for example, `name@example.com`)
- **targets** (*array*) – names of destination mailboxes or addresses
- **autoresponder_on** (*boolean*) – whether an autoresponder is enabled for the address (optional, default: false)
- **autoresponder_subject** (*string*) – subject line of the autoresponder message (optional, default: an empty string)
- **autoresponder_message** (*string*) – body of the autoresponder message (optional, default: an empty string)
- **autoresponder_from** (*string*) – originating address of the autoresponder message (optional, default: an empty string)
- **script_machine** (*string*) – a machine name for specifying a path to a script (optional, default: an empty string)
- **script_path** (*string*) – an absolute path to a script; see *Sending Mail to a Script* for details (optional, default: an empty string)

Change the details of an existing email address. The email address must exist before calling the method. The method returns a struct with the following key-value pairs:

id email ID

email_address email address

targets mailboxes or email addresses to which the address is set to deliver

See Also:

`create_email`

3.3 Websites and Domains

`create_domain`

Parameters

- **session_id** – session ID returned by `login`
- **domain** (*string*) – a domain name in the form of `example.com`
- **subdomain** (*string*) – each additional parameter provided after `domain`: a subdomain name of `domain`

Create a domain entry. If `domain` has already been created, you may supply additional parameters to add subdomains. For example, if `example.com` already exists, `create_domain` may be called with four parameters— a session ID, `example.com`, `www`, `private` —to create `www.example.com` and `private.example.com`.

Example: Create a domain entry for `widgetcompany.example` using Python:

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('widgetsco', 'widgetsrock')
>>> server.create_domain(session_id, 'widgetcompany.example', 'www', 'design')
{'domain': 'widgetcompany.example',
 'id': 47255,
 'subdomains': ['www', 'design']}
```

`create_website`

Parameters

- **session_id** – session ID returned by `login`
- **website_name** (*string*) – the name of the new website entry
- **ip** (*string*) – IP address of the server where the entry resides
- **https** (*boolean*) – whether the website entry should use a secure connection
- **subdomains** (*array*) – an array of strings of (sub)domains to be associated with the website entry
- **site_apps** (*array*) – each additional parameter provided after `subdomains`: an array containing a valid application name (a string) and a URL path (a string)

Create a new website entry. Applications may be added to the website entry with additional parameters supplied after `subdomains`. The additional parameters must be arrays containing two elements: a valid application name and a path (for example, `'htdocs'` and `'/'`).

Example: Create a website entry for `widgetcompany.example`'s new Django project over HTTPS using Python:

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('widgetsco', 'widgetsrock')
>>> server.create_website(session_id,
... 'widgets_on_the_web',
... '174.133.82.194',
... True,
... ['widgetcompany.example', 'www.widgetcompany.example'],
... ['django', '/'])
{'https': True,
 'id': 67074,
 'ip': '174.133.82.194',
 'name': 'widgets_on_the_web',
 'site_apps': [['django', '/']],
 'subdomains': ['widgetcompany.example', 'www.widgetcompany.example']}
```

`delete_domain`

Parameters

- **session_id** – session ID returned by `login`
- **domain** (*string*) – name of the domain to be deleted or the parent domain of the subdomains to be deleted
- **subdomains** (*string*) – each additional parameter provided after `domain`: subdomains of `domain` to be deleted

Delete a domain record or subdomain records. Subdomains of a domain may be deleted by supplying additional parameters after `domain`. If any subdomains are provided, only subdomains are deleted and the parent domain remains.

`delete_website`

Parameters

- **session_id** – session ID returned by `login`
- **website_name** (*string*) – name of website to be deleted
- **ip** (*string*) – IP address of the server where the website resides
- **https** (*boolean*) – whether the website uses a secure connection (optional, default: false)

Delete a website entry.

`list_bandwidth_usage`

Parameters **session_id** – session ID returned by `login`

List bandwidth usage statistics for your websites (similar to usage statistics shown on the control panel). The method returns a struct containing two members:

`daily`: A struct containing members named for the dates for the past two weeks (for example, `2015-01-05`, `2015-01-04`, `2015-01-03` and so on). The value of each dated member is a struct containing members named for each domain associated with the account (for example, `example.com`, `www.example.com`, `somedomain.example`, `www.somedomain.example` and so on). The value of each domain name member is the bandwidth usage for that domain during that day in kilobytes.

`monthly`: A struct containing members named for the months for the past year (for example, `2015-01`, `2014-12`, `2014-11` and so on). The value of each month member is a struct containing members named for each domain associated with the account (for example, `example.com`, `www.example.com`, `somedomain.example`, `www.somedomain.example` and so on). The value of each domain name member is the bandwidth usage for that domain during that month in kilobytes.

Overall, the struct resembles this outline:

- `daily`
  - –today
    - `www.example.com`: `1024`
    - `example.com`: `512`
    - *...
  - –yesterday
  - –...
  - –two weeks ago
- `monthly`
  - –this month
    - `www.example.com`: `2048`
    - `example.com`: `1024`
    - *...

–last month

–...

–a year ago

`list_domains`

Parameters **session_id** – session ID returned by `login`

Get information about the account's domains. The method returns an array of structs with the following key-value pairs:

id domain ID

domain domain (for example, `example.com`)

subdomains array of subdomains for the domain

`list_websites`

Parameters **session_id** – session ID returned by `login`

Get information about the account's websites. The method returns an array of structs with the following key-value pairs:

id website ID

name website name

ip website IP address

https whether the website is served over HTTPS

subdomains array of website's subdomains

website_apps array of the website's apps and their URL paths; each item in the array is a two-item array, containing an application name and URL path

`update_website`

Parameters

- **session_id** – session ID returned by `login`
- **website_name** (*string*) – the name of the website entry
- **ip** (*string*) – IP address of the server where the entry resides
- **https** (*boolean*) – whether the website entry should use a secure connection
- **subdomains** (*array*) – an array of strings of (sub)domains to be associated with the website entry
- **site_apps** (*array*) – each additional parameter provided after `subdomains`: an array containing a valid application name (a string) and a URL path (a string)

Update a website entry. Applications may be added to the website entry with additional parameters supplied after `subdomains`. The additional parameters must be arrays containing two elements: a valid application name and a path (for example, `'htdocs'` and `'/'`).

Example: Update a website entry for `widgetcompany.example`'s new Django project over HTTPS using Python:

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('widgetsco', 'widgetsrock')
>>> server.update_website(session_id,
... 'widgets_on_the_web',
... '174.133.82.195',
... True,
... ['widgetcompany.example', 'dev.widgetcompany.example'],
... ('django', '/'), ('wordpress', '/blog'))
{'https': True,
 'id': 67074,
 'ip': '174.133.82.195',
 'name': 'widgets_on_the_web',
 'site_apps': [['django', '/'], ['wordpress', '/blog']],
 'subdomains': ['widgetcompany.example', 'dev.widgetcompany.example']}
```

3.4 Applications

`create_app`

Parameters

- **session_id** – session ID returned by `login`
- **name** (*string*) – name of the application
- **type** (*string*) – type of the application
- **autostart** (*boolean*) – whether the app should restart with an `autostart.cgi` script (optional, default: false)
- **extra_info** (*string*) – additional information required by the application; if *extra_info* is not required or used by the application, it is ignored (optional, default: an empty string)
- **open_port** (*boolean*) – for applications that listen on a port, whether the port should be open on shared and dedicated IP addresses (optional, default: false)

Create a new application.

See Also:

For a complete list of application types, see *Application Types*.

`delete_app`

Parameters

- **session_id** – session ID returned by `login`
- **name** (*string*) – name of the application

Delete an application.

`list_apps`

Parameters **session_id** – session ID returned by `login`

Get information about the account's applications. The method returns an array of structs with the following key-value pairs:

id app ID

name app name

type app type

autostart whether the app uses autostart

port port number if the app listens on a port, otherwise is `0`

open_port for applications that listen on a port, whether the port is open on shared and dedicated IP addresses (`True` for open ports, `False` for closed ports, or for applications that do not listen to a port)

extra_info extra info for the app if any

machine name of the machine where the app resides

`list_app_types`

Parameters **session_id** – session ID returned by `login`

Get information about available app types. The method returns an array of structs with the following key-value pairs:

name an identifier for the application type (for use as the `create_app` method's `type` parameter)

label a short description of the application type

description a longer description of the application type

autostart `applicable` or an empty string, indicating whether the application uses an `autostart.cgi` file

extra_info description of any additional information required by the application installer's *extra_info* field

open_port a boolean value indicating whether the application may use an open port

See Also:

`create_app`

3.5 Cron

`create_cronjob`

Parameters

- **session_id** – session ID returned by `login`
- **line** (*string*) – crontab line to be added

Create a new cron job.

See Also:

For more information about the cron syntax, please see the Wikipedia entry on cron.

`delete_cronjob`

Parameters

- **session_id** – session ID returned by `login`
- **line** (*string*) – crontab line to be removed

Remove an existing cron job.

3.6 DNS

`create_dns_override`

Parameters

- **session_id** – session ID returned by `login`
- **domain** (*string*) – domain name to be overridden (for example, `sub.example.com`)
- **a_ip** (*string*) – A IP address (optional, default: an empty string)
- **cname** (*string*) – CNAME record (optional, default: an empty string)
- **mx_name** (*string*) – Mail exchanger record host name (optional, default: an empty string)
- **mx_priority** (*string*) – Mail exchanger record priority (optional, default: an empty string)
- **spf_record** (*string*) – TXT record (optional, default: an empty string)
- **aaaa_ip** (*string*) – An IPv6 address (optional, default: an empty string)

Create DNS records and return an array of the new records (as in the form of `list_dns_overrides`).

`delete_dns_override`

Parameters

- **session_id** – session ID returned by `login`
- **domain** (*string*) – domain name to be overridden (for example, `sub.example.com`)
- **a_ip** (*string*) – A IP address (optional, default: an empty string)
- **cname** (*string*) – CNAME record (optional, default: an empty string)
- **mx_name** (*string*) – Mail exchanger record host name (optional, default: an empty string)
- **mx_priority** (*string*) – Mail exchanger record priority (optional, default: an empty string)
- **spf_record** (*string*) – TXT record (optional, default: an empty string)
- **aaaa_ip** (*string*) – An IPv6 address (optional, default: an empty string)

Delete DNS records and return an array of the deleted records (as in the form of `list_dns_overrides`).

`list_dns_overrides`

Parameters **session_id** – session ID returned by `login`

Get information about the account's DNS overrides. The method returns an array of structs with the following key-value pairs:

id domain ID

domain domain name to be overridden (for example, `sub.example.com`)

a_ip A IP address

aaaa_ip AAAA IP address (for IPv6)

cname CNAME record

mx_name Mail exchanger record host name

mx_priority Mail exchanger record priority

spf_record TXT record

srv_record Service record

3.7 Databases

`change_db_user_password`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – a database user's username
- **password** (*string*) – the new password
- **db_type** (*string*) – the database type, either `mysql` or `postgresql`

Change a database user's password. The method returns a struct containing the following key-value pairs:

username database username

machine database machine name

db_type database type (MySQL or PostgreSQL)

database database name

See Also:

See Strengthening Passwords for important information about choosing passwords.

`create_db`

Parameters

- **session_id** – session ID returned by `login`
- **name** (*string*) – database name
- **db_type** (*string*) – the database type, either `mysql` or `postgresql`
- **password** (*string*) – password for the default database user

Create a database.

Note: MySQL database names may not exceed 16 characters.

See Also:

See Strengthening Passwords for important information about choosing passwords.

`create_db_user`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – the new database user's username
- **password** (*string*) – the new database user's password
- **db_type** (*string*) – the database type, either `mysql` or `postgresql`

Create a database user. The method returns a struct with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

See Also:

See Strengthening Passwords for important information about choosing passwords.

`delete_db`

Parameters

- **session_id** – session ID returned by `login`
- **name** (*string*) – database name
- **db_type** (*string*) – the database type, either `mysql` or `postgresql`

Delete a database.

`delete_db_user`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – the database user's username
- **db_type** (strings `mysql` or `postgresql`) – the database type

Delete a database user. The method returns a struct with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

`enable_addon`

Parameters

- **session_id** – session ID returned by `login`
- **database** (*string*) – a database name
- **db_type** (*string*) – the database type (always use `postgresql`)
- **addon** (*string*) – the addon to enable (`tsearch` or `postgis`)

Enable a database addon. The method returns a struct with the following key-value pairs:

machine machine name

db_type database type (always PostgreSQL)

addon addon enabled

db_type database type (MySQL or PostgreSQL)

Note: This method applies to PostgreSQL databases only.

`grant_db_permissions`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – a database user's username

- **database** (*string*) – a database name
- **db_type** (*string*) – the database type (`mysql` or `postgresql`)

Grant full database permissions to a user with respect to a database. The method returns a struct with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

database database name

`list_dbs`

Parameters **session_id** – session ID returned by `login`

Get information about the account's databases. The method returns an array of structs with the following key-value pairs:

id database ID

name database name

db_type database type (MySQL or PostgreSQL)

machine machine name

`list_db_users`

Parameters **session_id** – session ID returned by `login`

Get information about the account's database users. The method returns an array of structs with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

`make_user_owner_of_db`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – a database user's username
- **database** (*string*) – a database name
- **db_type** (*string*) – the database type (`mysql` or `postgresql`)

Assign ownership of a database to a user. The method returns a struct with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

database database name

`revoke_db_permissions`

Parameters

- **session_id** – session ID returned by `login`

- **username** (*string*) – a database user's username
- **database** (*string*) – a database name
- **db_type** (*string*) – the database type (`mysql` or `postgresql`)

Revoke a user's permissions with respect to a database. The method returns a struct with the following key-value pairs:

machine machine name

username database username

db_type database type (MySQL or PostgreSQL)

database database name

3.8 Files

`replace_in_file`

Parameters

- **session_id** – session ID returned by `login`
- **filename** (*string*) – path to file from the user's home directory
- **changes** (*array*) – each additional parameter provided after `filename` : an array of two strings, where the first is the text to be replaced and the second is the replacement text

Find and replace strings in a file in the users's home directory tree.

Any parameters after `filename` must be arrays containing a pair of strings, where the first is the string to be replaced and the second is the replacement text.

Example: Find all appearances of the word "eggs" in a file in the user's home directory and replace them with the word "spam" using Python:

```
$ cat myfile.txt
eggs, spam, spam, and spam.
spam, spam, spam, and eggs.
```

```
>>> import xmlrpclib
>>> server = xmlrpclib.ServerProxy('https://api.webfaction.com/')
>>> session_id, account = server.login('widgetsco', 'widgetsrock')
>>> server.replace_in_file(session_id, 'myfile.txt', ('eggs', 'spam'))
''
>>> exit()
```

```
$ cat myfile.txt
spam, spam, spam, and spam.
spam, spam, spam, and spam.
```

`replace_in_file`

`write_file`

Parameters

- **session_id** – session ID returned by `login`
- **filename** (*string*) – path to file to be written from the user's home directory

- **str** (*string*) – string to be written
- **mode** (*string*) – write mode (optional, default: `wb`)

Write a string to a file in the user's home directory tree.

Note: Commonly, the write mode is `w` for plain text and `wb` for binaries. `a` and `ab` can be used to append to files.

See Also:

For more information about write modes, please see open().

3.9 Shell Users

`change_user_password`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – username
- **password** (*string*) – a new password

Change a shell user's password.

See Also:

See Strengthening Passwords for important information about choosing passwords.

`create_user`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – username
- **shell** (*string*) – the user's command line interpreter; one of `none`, `bash`, `sh`, `ksh`, `csh`, or `tcsh`.
- **groups** (*array*) – extra permission groups of which the new user is to be a member

Create a new shell user. If `shell` is `none`, the user has FTP access only. All users are automatically a member of their own group; do not include the user's own group in `groups`. Use an empty array to specify no extra groups.

`delete_user`

Parameters

- **session_id** – session ID returned by `login`
- **username** (*string*) – username

Delete a user.

`list_users`

Parameters **session_id** – session ID returned by `login`

Get information about the account's shell users. The method returns an array of structs with the following key-value pairs:

username username

machine the user's server (for example, `Web100`)

shell the user's configured command line interpreter (for example, `bash` or `tcsh`)

groups extra permissions groups of which the user is a member

3.10 Servers

`list_ips`

Parameters **session_id** – session ID returned by `login`

Get information about all of the account's machines and their IP addresses. This method returns an array of structs with the following key-value pairs:

machine machine name (for example, `Web100`)

ip IP address

is_main a boolean value indicating whether the IP address is the primary address for the server (true) or an extra IP address provisioned to the account (false)

`list_machines`

Parameters **session_id** – session ID returned by `login`

Get information about the account's machines. This method returns an array of structs with the following key-value pairs:

name machine name (for example, `Web100`)

operating_system the machine's operating system (for example, `Centos6-64bit`)

location the machine's location (for example, `USA`)

3.11 Miscellaneous

`run_php_script`

Parameters

- **session_id** – session ID returned by `login`
- **script** (*string*) – an absolute path to script (or path to the script starting from the user's home directory)
- **code_before** (*string*) – PHP code to be executed before `script`

Run PHP code followed by a PHP script. The PHP code and script is run as the user.

`set_apache_acl`

Parameters

- **session_id** – session ID returned by `login`
- **paths** (*string or array of strings*) – path from home directory
- **permission** (*string*) – `r`, `w`, or `x` or a combination thereof (optional, default: `rwx`)

- **recursive** (*boolean*) – whether Apache's access is granted recursively (optional, default: false)

Grant the machine-wide Apache instance access to files or directories.

`system`

Parameters

- **session_id** – session ID returned by `login`
- **cmd** (*string*) – a shell command to be executed

Execute a command as the user, as if through SSH. If an application was installed previously in the session, the command will be run from the directory where that application was installed.

Note: If `cmd` writes to standard error, the API method will return an XML-RPC fault.

CHAPTER

FOUR

APPLICATION TYPES

The following application types may be used with API methods such as `create_app`. Each entry contains the application type's unique name paired with it's descriptive label.

awstats66 AWStats (6.6) - INSECURE

awstats68 AWStats (6.8) - INSECURE

awstats70 AWStats (7.0) - INSECURE

awstats73 AWStats (7.3)

cherrypy210_24 CherryPy 2.1.0 (Python 2.4)

cherrypy221_24 CherryPy 2.2.1 (Python 2.4)

cherrypy300_24 CherryPy 3.0.0 (Python 2.4)

cherrypy310_24 CherryPy 3.1.0 (Python 2.4)

cherrypy312_25 CherryPy 3.1.2 (Python 2.5)

cherrypy320_25 CherryPy 3.2.0 (Python 2.5)

cherrypy320_26 CherryPy 3.2.0 (Python 2.6)

cherrypy320_27 CherryPy 3.2.0 (Python 2.7)

cherrypy320_31 CherryPy 3.2.0 (Python 3.1)

cherrypy322_27 CherryPy 3.2.2 (Python 2.7)

cherrypy322_32 CherryPy 3.2.2 (Python 3.2)

cherrypy324_27 CherryPy 3.2.4 (Python 2.7)

cherrypy324_32 CherryPy 3.2.4 (Python 3.2)

cherrypy324_34 CherryPy 3.2.4 (Python 3.4)

cherrypy360_27 CherryPy 3.6.0 (Python 2.7)

cherrypy360_34 CherryPy 3.6.0 (Python 3.4)

custom_app_with_port Custom app (listening on port)

custom_install_script Custom install script

`custom_websockets_app_with_port` Custom websockets app (listening on port)

`django091_mp331_24` Django 0.91 (mod_python 3.3.1/Python 2.4) - DISCONTINUED

`django0951_mp331_24` Django 0.95.1 (mod_python 3.3.1/Python 2.4) - DISCONTINUED

`django095_mp331_24` Django 0.95 (mod_python 3.3.1/Python 2.4) - DISCONTINUED

`django0962_mp331_25` Django 0.96.2 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django0962_mw20_25` Django 0.96.2 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django0963_mp331_25` Django 0.96.3 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django0963_mw20_25` Django 0.96.3 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django0964_mp331_25` Django 0.96.4 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django096_mp331_24` Django 0.96 (mod_python 3.3.1/Python 2.4) - DISCONTINUED

`django096_mp331_25` Django 0.96 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django096_mw20_25` Django 0.96 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django101_mp331_25` Django 1.0.1 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django101_mw20_25` Django 1.0.1 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django102_mp331_25` Django 1.0.2 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django102_mw20_25` Django 1.0.2 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django103_mp331_25` Django 1.0.3 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django104_mp331_25` Django 1.0.4 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django104_mw25_25` Django 1.0.4 (mod_wsgi 2.5/Python 2.5)- DISCONTINUED

`django10_mp331_25` Django 1.0 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django10_mw20_25` Django 1.0 (mod_wsgi 2.0/Python 2.5) - INSECURE

`django111_mp331_25` Django 1.1.1 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django111_mw25_25` Django 1.1.1 (mod_wsgi 2.5/Python 2.5) - INSECURE

`django112_mw32_25` Django 1.1.2 (mod_wsgi 3.2/Python 2.5) INSECURE

`django112_mw32_26` Django 1.1.2 (mod_wsgi 3.2/Python 2.6) INSECURE

`django113_mw32_26` Django 1.1.3 (mod_wsgi 3.2/Python 2.6) - INSECURE

`django114_mw32_26` Django 1.1.4 (mod_wsgi 3.2/Python 2.6) - INSECURE

`django11_mp331_25` Django 1.1 (mod_python 3.3.1/Python 2.5) - DISCONTINUED

`django11_mw25_25` Django 1.1 (mod_wsgi 2.5/Python 2.5) - INSECURE

`django121_mw32_25` Django 1.2.1 (mod_wsgi 3.2/Python 2.5) - INSECURE

`django121_mw32_26` Django 1.2.1 (mod_wsgi 3.2/Python 2.6) - INSECURE

django122_mw32_26 Django 1.2.2 (mod_wsgi 3.2/Python 2.6) - INSECURE

django123_mw32_26 Django 1.2.3 (mod_wsgi 3.2/Python 2.6) - INSECURE

django124_mw32_26 Django 1.2.4 (mod_wsgi 3.2/Python 2.6) - INSECURE

django125_mw32_26 Django 1.2.5 (mod_wsgi 3.2/Python 2.6) - INSECURE

django125_mw33_27 Django 1.2.5 (mod_wsgi 3.3/Python 2.7) - INSECURE

django126_mw32_26 Django 1.2.6 (mod_wsgi 3.2/Python 2.6) - INSECURE

django126_mw33_27 Django 1.2.6 (mod_wsgi 3.3/Python 2.7) - INSECURE

django127_mw32_26 Django 1.2.7 (mod_wsgi 3.2/Python 2.6) - INSECURE

django127_mw33_27 Django 1.2.7 (mod_wsgi 3.3/Python 2.7) - INSECURE

django131_mw32_26 Django 1.3.1 (mod_wsgi 3.2/Python 2.6) - INSECURE

django131_mw33_27 Django 1.3.1 (mod_wsgi 3.3/Python 2.7) - INSECURE

django132_mw32_26 Django 1.3.2 (mod_wsgi 3.2/Python 2.6) - INSECURE

django132_mw33_27 Django 1.3.2 (mod_wsgi 3.3/Python 2.7) - INSECURE

django133_mw32_26 Django 1.3.3 (mod_wsgi 3.2/Python 2.6) - INSECURE

django133_mw33_27 Django 1.3.3 (mod_wsgi 3.3/Python 2.7) - INSECURE

django134_mw32_26 Django 1.3.4 (mod_wsgi 3.2/Python 2.6) - INSECURE

django134_mw33_27 Django 1.3.4 (mod_wsgi 3.3/Python 2.7) - INSECURE

django135_mw32_26 Django 1.3.5 (mod_wsgi 3.2/Python 2.6) - INSECURE

django135_mw33_27 Django 1.3.5 (mod_wsgi 3.3/Python 2.7) - INSECURE

django136_mw34_27 Django 1.3.6 (mod_wsgi 3.4/Python 2.7) - INSECURE

django137_mw34_27 Django 1.3.7 (mod_wsgi 3.4/Python 2.7) - INSECURE

django13_mw32_26 Django 1.3 (mod_wsgi 3.2/Python 2.6) - INSECURE

django13_mw33_27 Django 1.3 (mod_wsgi 3.3/Python 2.7) - INSECURE

django1411_mw34_27 Django 1.4.11 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1412_mw34_27 Django 1.4.12 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1413_mw34_27 Django 1.4.13 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1414_mw34_27 Django 1.4.14 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1415_mw35_27 Django 1.4.15 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1416_mw35_27 Django 1.4.16 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1417_mw35_27 Django 1.4.17 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1418_mw35_27 Django 1.4.18 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1419_mw35_27 Django 1.4.19 (mod_wsgi 3.5/Python 2.7) - INSECURE

django141_mw33_27 Django 1.4.1 (mod_wsgi 3.3/Python 2.7) - INSECURE

django141_mw34_27 Django 1.4.1 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1420_mw35_27 Django 1.4.20 (mod_wsgi 3.5/Python 2.7)

django142_mw33_27 Django 1.4.2 (mod_wsgi 3.3/Python 2.7) - INSECURE

django142_mw34_27 Django 1.4.2 (mod_wsgi 3.4/Python 2.7) - INSECURE

django143_mw33_27 Django 1.4.3 (mod_wsgi 3.3/Python 2.7) - INSECURE

django143_mw34_27 Django 1.4.3 (mod_wsgi 3.4/Python 2.7) - INSECURE

django144_mw34_27 Django 1.4.4 (mod_wsgi 3.4/Python 2.7) - INSECURE

django145_mw34_27 Django 1.4.5 (mod_wsgi 3.4/Python 2.7) - INSECURE

django146_mw34_27 Django 1.4.6 (mod_wsgi 3.4/Python 2.7) - INSECURE

django147_mw34_27 Django 1.4.7 (mod_wsgi 3.4/Python 2.7) - INSECURE

django148_mw34_27 Django 1.4.8 (mod_wsgi 3.4/Python 2.7) - INSECURE

django149_mw34_27 Django 1.4.9 (mod_wsgi 3.4/Python 2.7) - INSECURE

django14_mw33_27 Django 1.4 (mod_wsgi 3.3/Python 2.7) - INSECURE

django1510_mw35_27 Django 1.5.10 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1510_mw35_34 Django 1.5.10 (mod_wsgi 3.5/Python 3.4) - INSECURE

django1511_mw35_27 Django 1.5.11 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1511_mw35_34 Django 1.5.11 (mod_wsgi 3.5/Python 3.4) - INSECURE

django1512_mw35_27 Django 1.5.12 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1512_mw35_34 Django 1.5.12 (mod_wsgi 3.5/Python 3.4) - INSECURE

django151_mw34_27 Django 1.5.1 (mod_wsgi 3.4/Python 2.7) - INSECURE

django152_mw34_27 Django 1.5.2 (mod_wsgi 3.4/Python 2.7) - INSECURE

django153_mw34_27 Django 1.5.3 (mod_wsgi 3.4/Python 2.7) - INSECURE

django154_mw34_27 Django 1.5.4 (mod_wsgi 3.4/Python 2.7) - INSECURE

django154_mw34_33 Django 1.5.4 (mod_wsgi 3.4/Python 3.3) - INSECURE

django155_mw34_27 Django 1.5.5 (mod_wsgi 3.4/Python 2.7) - INSECURE

django155_mw34_33 Django 1.5.5 (mod_wsgi 3.4/Python 3.3) - INSECURE

django156_mw34_27 Django 1.5.6 (mod_wsgi 3.4/Python 2.7) - INSECURE

django156_mw34_33 Django 1.5.6 (mod_wsgi 3.4/Python 3.3) - INSECURE

django157_mw34_27 Django 1.5.7 (mod_wsgi 3.4/Python 2.7) - INSECURE

django157_mw34_33 Django 1.5.7 (mod_wsgi 3.4/Python 3.3) - INSECURE

django158_mw34_27 Django 1.5.8 (mod_wsgi 3.4/Python 2.7) - INSECURE

django158_mw34_33 Django 1.5.8 (mod_wsgi 3.4/Python 3.3) - INSECURE

django159_mw34_27 Django 1.5.9 (mod_wsgi 3.4/Python 2.7) - INSECURE

django159_mw34_33 Django 1.5.9 (mod_wsgi 3.4/Python 3.3) - INSECURE

django15_mw34_27 Django 1.5 (mod_wsgi 3.4/Python 2.7) - INSECURE

django1610_mw35_27 Django 1.6.10 (mod_wsgi 3.5/Python 2.7) - INSECURE

django1610_mw35_34 Django 1.6.10 (mod_wsgi 3.5/Python 3.4) - INSECURE

django1611_mw35_27 Django 1.6.11 (mod_wsgi 3.5/Python 2.7)

django1611_mw35_34 Django 1.6.11 (mod_wsgi 3.5/Python 3.4)

django161_mw34_27 Django 1.6.1 (mod_wsgi 3.4/Python 2.7) - INSECURE

django161_mw34_33 Django 1.6.1 (mod_wsgi 3.4/Python 3.3) - INSECURE

django162_mw34_27 Django 1.6.2 (mod_wsgi 3.4/Python 2.7) - INSECURE

django162_mw34_33 Django 1.6.2 (mod_wsgi 3.4/Python 3.3) - INSECURE

django163_mw34_27 Django 1.6.3 (mod_wsgi 3.4/Python 2.7) - INSECURE

django163_mw34_33 Django 1.6.3 (mod_wsgi 3.4/Python 3.3) - INSECURE

django164_mw34_27 Django 1.6.4 (mod_wsgi 3.4/Python 2.7) - INSECURE

django164_mw34_33 Django 1.6.4 (mod_wsgi 3.4/Python 3.3) - INSECURE

django165_mw34_27 Django 1.6.5 (mod_wsgi 3.4/Python 2.7) - INSECURE

django165_mw34_33 Django 1.6.5 (mod_wsgi 3.4/Python 3.3) - INSECURE

django165_mw34_34 Django 1.6.5 (mod_wsgi 3.4/Python 3.4) - INSECURE

django166_mw34_27 Django 1.6.6 (mod_wsgi 3.4/Python 2.7) - INSECURE

django166_mw34_33 Django 1.6.6 (mod_wsgi 3.4/Python 3.3) - INSECURE

django166_mw34_34 Django 1.6.6 (mod_wsgi 3.4/Python 3.4) - INSECURE

django167_mw35_27 Django 1.6.7 (mod_wsgi 3.5/Python 2.7) - INSECURE

django167_mw35_34 Django 1.6.7 (mod_wsgi 3.5/Python 3.4) - INSECURE

django168_mw35_27 Django 1.6.8 (mod_wsgi 3.5/Python 2.7) - INSECURE

django168_mw35_34 Django 1.6.8 (mod_wsgi 3.5/Python 3.4) - INSECURE

django169_mw35_27 Django 1.6.9 (mod_wsgi 3.5/Python 2.7) - INSECURE

django169_mw35_34 Django 1.6.9 (mod_wsgi 3.5/Python 3.4) - INSECURE

django16_mw34_27 Django 1.6 (mod_wsgi 3.4/Python 2.7) - INSECURE

django16_mw34_33 Django 1.6 (mod_wsgi 3.4/Python 3.3) - INSECURE

django171_mw35_27 Django 1.7.1 (mod_wsgi 3.5/Python 2.7) - INSECURE

django171_mw35_34 Django 1.7.1 (mod_wsgi 3.5/Python 3.4) - INSECURE

django172_mw35_27 Django 1.7.2 (mod_wsgi 3.5/Python 2.7) - INSECURE

django172_mw35_34 Django 1.7.2 (mod_wsgi 3.5/Python 3.4) - INSECURE

django173_mw35_27 Django 1.7.3 (mod_wsgi 3.5/Python 2.7) - INSECURE

django173_mw35_34 Django 1.7.3 (mod_wsgi 3.5/Python 3.4) - INSECURE

django174_mw35_27 Django 1.7.4 (mod_wsgi 3.5/Python 2.7) - INSECURE

django174_mw35_34 Django 1.7.4 (mod_wsgi 3.5/Python 3.4) - INSECURE

django176_mw35_27 Django 1.7.6 (mod_wsgi 3.5/Python 2.7) - INSECURE

django176_mw35_34 Django 1.7.6 (mod_wsgi 3.5/Python 3.4) - INSECURE

django177_mw35_27 Django 1.7.7 (mod_wsgi 3.5/Python 2.7) - INSECURE

django177_mw35_34 Django 1.7.7 (mod_wsgi 3.5/Python 3.4) - INSECURE

django178_mw4411_27 Django 1.7.8 (mod_wsgi 4.4.11/Python 2.7)

django178_mw4411_34 Django 1.7.8 (mod_wsgi 4.4.11/Python 3.4)

django17_mw35_27 Django 1.7 (mod_wsgi 3.5/Python 2.7) - INSECURE

django17_mw35_34 Django 1.7 (mod_wsgi 3.5/Python 3.4) - INSECURE

django181_mw4411_27 Django 1.8.1 (mod_wsgi 4.4.11/Python 2.7)

django181_mw4411_34 Django 1.8.1 (mod_wsgi 4.4.11/Python 3.4)

django18_mw428_27 Django 1.8 (mod_wsgi 4.2.8/Python 2.7)

django18_mw428_34 Django 1.8 (mod_wsgi 4.2.8/Python 3.4)

djangotrunk_mp331_25 Django (trunk)/mod_python (3.3.1)/Python (2.5) - DISCONTINUED

djangotrunk_mw20_25 Django (trunk)/mod_wsgi (2.0)/Python (2.5) - DISCONTINUED

djangotrunk_mw25_25 Django (trunk)/mod_wsgi (2.5)/Python (2.5) - DISCONTINUED

djangotrunk_mw33_27 Django (trunk)/mod_wsgi (3.3)/Python (2.7)

drupal_5_1 Drupal (5.1) - INSECURE

drupal_5_2 Drupal (5.2) - INSECURE

drupal_5_7 Drupal (5.7) - INSECURE

drupal_6_1 Drupal (6.1) - INSECURE

drupal_6_10 Drupal (6.10) - INSECURE

drupal_6_12 Drupal (6.12) - INSECURE

`drupal_6_13` Drupal (6.13) - INSECURE

`drupal_6_14` Drupal (6.14) - INSECURE

`drupal_6_15` Drupal (6.15) - INSECURE

`drupal_6_17` Drupal (6.17) - INSECURE

`drupal_6_19` Drupal (6.19) - INSECURE

`drupal_6_2` Drupal (6.2) - INSECURE

`drupal_6_20` Drupal (6.20) - INSECURE

`drupal_6_22` Drupal (6.22) - INSECURE

`drupal_6_28` Drupal (6.28) - INSECURE

`drupal_6_32` Drupal (6.32) - INSECURE

`drupal_6_33` Drupal (6.33) - INSECURE

`drupal_6_34` Drupal (6.34) - INSECURE

`drupal_6_35` Drupal (6.35)

`drupal_6_4` Drupal (6.4) - INSECURE

`drupal_6_6` Drupal (6.6) - INSECURE

`drupal_7` Drupal (7.0) - INSECURE

`drupal_7_12` Drupal (7.12) - INSECURE

`drupal_7_14` Drupal (7.14) - INSECURE

`drupal_7_15` Drupal (7.15) - INSECURE

`drupal_7_16` Drupal (7.16) - INSECURE

`drupal_7_18` Drupal (7.18) - INSECURE

`drupal_7_19` Drupal (7.19) - INSECURE

`drupal_7_2` Drupal (7.2) - INSECURE

`drupal_7_20` Drupal (7.20) - INSECURE

`drupal_7_21` Drupal (7.21) - INSECURE

`drupal_7_23` Drupal (7.23) - INSECURE

`drupal_7_24` Drupal (7.24) - INSECURE

`drupal_7_26` Drupal (7.26) - INSECURE

`drupal_7_27` Drupal (7.27) - INSECURE

`drupal_7_29` Drupal (7.29) - INSECURE

`drupal_7_30` Drupal (7.30) - INSECURE

`drupal_7_31` Drupal (7.31) - INSECURE

`drupal_7_32` Drupal (7.32) - INSECURE

`drupal_7_34` Drupal (7.34) - INSECURE

`drupal_7_35` Drupal (7.35)

`drupal_7_4` Drupal (7.4) - INSECURE

`drupal_7_7` Drupal (7.7) - INSECURE

`drupal_7_8` Drupal (7.8) - INSECURE

`drupal_7_9` Drupal (7.9) - INSECURE

`ghost-0.3.3` Ghost 0.3.3

`ghost-0.4.0` Ghost 0.4.0

`ghost-0.5.0` Ghost 0.5.0

`ghost-0.5.5` Ghost 0.5.5

`ghost-0.5.8` Ghost 0.5.8

`ghost-0.6.0` Ghost 0.6.0

`git` Git 1.7.4.1 - DISCONTINUED

`git_171` Git 1.7.1 - DISCONTINUED

`git_172` Git 1.7.2 - DISCONTINUED

`git_193` Git 1.9.3 - DISCONTINUED

`git_230` Git 2.3.0

`joomla_150` Joomla (1.5.0) - DISCONTINUED

`joomla_1510` Joomla (1.5.10) - DISCONTINUED

`joomla_1515` Joomla (1.5.15) - DISCONTINUED

`joomla_1520` Joomla (1.5.20) - DISCONTINUED

`joomla_157` Joomla (1.5.7) - DISCONTINUED

`joomla_159` Joomla (1.5.9) - DISCONTINUED

`joomla_16` Joomla (1.6.0) - DISCONTINUED

`joomla_161` Joomla (1.6.1) - DISCONTINUED

`joomla_170` Joomla (1.7.0) - INSECURE

`joomla_250` Joomla (2.5.0) - INSECURE

`joomla_251` Joomla (2.5.1) - INSECURE

`joomla_258` Joomla (2.5.8) - INSECURE

joomla_302 Joomla (3.0.2) - INSECURE

joomla_315 Joomla (3.1.5) - INSECURE

joomla_322 Joomla (3.2.2) - INSECURE

joomla_336 Joomla (3.3.6)

joomla_340 Joomla (3.4.0)

mod_python3210_24 mod_python (3.2.10)/Python (2.4) - DISCONTINUED

mod_python331_24 mod_python (3.3.1)/Python (2.4) - DISCONTINUED

mod_python331_25 mod_python (3.3.1)/Python (2.5) - DISCONTINUED

mod_wsgi25_25 mod_wsgi 2.5/Python 2.5 - DISCONTINUED

mod_wsgi2_25 mod_wsgi 2.0/Python 2.5 - DISCONTINUED

mod_wsgi32-python25 mod_wsgi 3.2/Python 2.5 - INSECURE

mod_wsgi32-python26 mod_wsgi 3.2/Python 2.6 - INSECURE

mod_wsgi33-python26 mod_wsgi 3.3/Python 2.6 - INSECURE

mod_wsgi33-python27 mod_wsgi 3.3/Python 2.7 - INSECURE

mod_wsgi33-python31 mod_wsgi 3.3/Python 3.1 - INSECURE

mod_wsgi34-python27 mod_wsgi 3.4/Python 2.7 - INSECURE

mod_wsgi34-python32 mod_wsgi 3.4/Python 3.2 - INSECURE

mod_wsgi34-python33 mod_wsgi 3.4/Python 3.3 - INSECURE

mod_wsgi34-python34 mod_wsgi 3.4/Python 3.4 - INSECURE

mod_wsgi35-python27 mod_wsgi 3.5/Python 2.7

mod_wsgi35-python32 mod_wsgi 3.5/Python 3.2

mod_wsgi35-python33 mod_wsgi 3.5/Python 3.3

mod_wsgi35-python34 mod_wsgi 3.5/Python 3.4

mod_wsgi428-python27 mod_wsgi 4.2.8/Python 2.7

mod_wsgi428-python34 mod_wsgi 4.2.8/Python 3.4

mod_wsgi4411-python27 mod_wsgi 4.4.11/Python 2.7

mod_wsgi4411-python34 mod_wsgi 4.4.11/Python 3.4

mysql MySQL private instance

node-0.10.21 Node.js 0.10.21

node-0.10.24 Node.js 0.10.24

node-0.10.31 Node.js 0.10.31

node-0.12.0 Node.js 0.12.0

passenger-2.2.15 Passenger 2.2.15 (nginx 0.7.64/Ruby Enterprise Edition 1.8.7)

passenger-2.2.8 Passenger 2.2.8 (nginx 0.7.64/Ruby Enterprise Edition 1.8.7)

passenger-3.0.11 Passenger 3.0.11 (nginx 1.0.10/Ruby 1.9.3)

passenger-3.0.3 Passenger 3.0.3 (nginx 0.8.54/Ruby Enterprise Edition 1.8.7)

passenger-3.0.9 Passenger 3.0.9 (nginx 1.0.6/Ruby Enterprise Edition 1.8.7)

passenger-4.0.2 Passenger 4.0.2 (nginx 1.4.1/Ruby 2.0.0)

passenger-4.0.41 Passenger 4.0.41 (nginx 1.4.7/Ruby 2.0)

passenger-4.0.53 Passenger 4.0.53 (nginx 1.6.2/Ruby 2.1)

passenger-5.0.4 Passenger 5.0.4 (nginx 1.6.2/Ruby 2.2)

passenger-5.0.6 Passenger 5.0.6 (nginx 1.6.2/Ruby 2.2)

plone335_zope21011 Zope (2.10.11) - Plone (3.3.5) - DISCONTINUED

plone401_zope21211 Zope (2.12.11) - Plone (4.0.1) - INSECURE

plone402_zope21213 Zope (2.12.13) - Plone (4.0.2) - INSECURE

plone404_zope21216 Zope (2.12.16) - Plone (4.0.4) - INSECURE

plone405_zope21217 Zope (2.12.17) - Plone (4.0.5)

plone40_zope21210 Zope (2.12.10) - Plone (4.0) - INSECURE

plone412_zope21310 Zope (2.13.10) - Plone (4.1.2)

plone416_zope21315 Zope (2.13.15) - Plone (4.1.6)

plone41_zope21214 Zope (2.12.14) - Plone (4.1)

plone423_zope21319 Zope (2.13.19) - Plone (4.2.3)

plone42_zope21315 Zope (2.13.15) - Plone (4.2)

plone433_zope21322 Zope (2.13.22) - Plone (4.3.3)

plone43_zope21319 Zope (2.13.19) - Plone (4.3)

postgresql PostgreSQL private instance

pylons_095 Pylons 0.9.5/Python 2.4 - INSECURE

pylons_0961 Pylons 0.9.6.1/Python 2.5 - INSECURE

pylons_0962 Pylons 0.9.6.2/Python 2.5 - DISCONTINUED

pylons_097 Pylons 0.9.7/Python 2.5 - DISCONTINUED

pylons_10_25 Pylons 1.0/Python 2.5 - DISCONTINUED

pylons_10_26 Pylons 1.0/Python 2.6 - DISCONTINUED

pyramid_11_27 Pyramid 1.1/Python 2.7

pyramid_12_27 Pyramid 1.2/Python 2.7

pyramid_132_27 Pyramid 1.3.2/Python 2.7

pyramid_13_27 Pyramid 1.3/Python 2.7

pyramid_14_27 Pyramid 1.4/Python 2.7

pyramid_15_27 Pyramid 1.5/Python 2.7

rails-2.3.14 Rails 2.3.14 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-2.3.15 Rails 2.3.15 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-2.3.16 Rails 2.3.16 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-2.3.17 Rails 2.3.17 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-2.3.18 Rails 2.3.18 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7)

rails-2.3.5 Rails 2.3.5 (nginx 0.7.64/Passenger 2.2.8/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-2.3.8 Rails 2.3.8 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.0.0 Rails 3.0.0 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.0.1 Rails 3.0.1 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.0.19 Rails 3.0.19 (nginx 0.8.54/Passenger 3.0.3/Ruby 1.8.7) - INSECURE

rails-3.0.20 Rails 3.0.20 (nginx 0.8.54/Passenger 3.0.3/Ruby 1.8.7)

rails-3.0.3 Rails 3.0.3 (nginx 0.7.64/Passenger 2.2.15/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.0.4 Rails 3.0.4 (nginx 0.8.54/Passenger 3.0.3/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.0.5 Rails 3.0.5 (nginx 0.8.54/Passenger 3.0.3/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.1.0 Rails 3.1.0 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.1.10 Rails 3.1.10 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.1.11 Rails 3.1.11 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7) - INSECURE

rails-3.1.12 Rails 3.1.12 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

rails-3.2.1 Rails 3.2.1 (nginx 1.0.10/Passenger 3.0.11/Ruby 1.9.3) - INSECURE

rails-3.2.11 Rails 3.2.11 (nginx 1.0.10/Passenger 3.0.11/Ruby 1.9.3) - INSECURE

rails-3.2.12 Rails 3.2.12 (nginx 1.0.10/Passenger 3.0.11/Ruby 1.9.3) - INSECURE

rails-3.2.13 Rails 3.2.13(nginx 1.0.10/Passenger 3.0.11/Ruby 1.9.3)

rails-3.2.13-ruby2 Rails 3.2.13(nginx 1.4.1/Passenger 4.0.2/Ruby 2.0.0)

rails-3.2.3 Rails 3.2.3 (nginx 1.0.10/Passenger 3.0.11/Ruby 1.9.3) - INSECURE

rails-4.0.0 Rails 4.0.0 (nginx 1.4.1/Passenger 4.0.2/Ruby 2.0.0)

rails-4.0.4 Rails 4.0.4 (nginx 1.4.1/Passenger 4.0.2/Ruby 2.0.0)

rails-4.1.1 Rails 4.1.1 (nginx 1.4.1/Passenger 4.0.2/Ruby 2.0.0)

rails-4.1.8 Rails 4.1.8 (nginx 1.6.2/Passenger 4.0.53/Ruby 2.1.2)

rails-4.2.0 Rails 4.2.0 (nginx 1.6.2/Passenger 4.0.53/Ruby 2.1.2)

rails-4.2.1-22 Rails 4.2.1 (nginx 1.6.2/Passenger 5.0.6/Ruby 2.2)

rails_112 Rails (1.1.2) - INSECURE

rails_116 Rails (1.1.6) - INSECURE

rails_121 Rails (1.2.1) - INSECURE

rails_123 Rails (1.2.3) - INSECURE

rails_201 Rails (2.0.1) - INSECURE

rails_202 Rails (2.0.2) - INSECURE

rails_210 Rails (2.1.0) - INSECURE

redmine-1.2.1 Redmine 1.2.1 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-1.2.2 Redmine 1.2.2 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-1.3.0 Redmine 1.3.0 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-1.3.1 Redmine 1.3.1 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-1.4.0 Redmine 1.4.0 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-1.4.1 Redmine 1.4.1 (nginx 1.0.6/Passenger 3.0.9/Ruby Enterprise Edition 1.8.7)

redmine-2.0.3 Redmine 2.0.3 - INSECURE

redmine-2.2.0 Redmine 2.2.0

redmine-2.2.2 Redmine 2.2.2 - INSECURE

redmine-2.2.4 Redmine 2.2.4

redmine-2.3.0 Redmine 2.3.0

redmine-2.3.1 Redmine 2.3.1

redmine-2.4.2 Redmine 2.4.2

redmine-2.6.0 Redmine 2.6.0

static Static/CGI/PHP-5.2

static_only Static only (no .htaccess)

static_php53 Static/CGI/PHP-5.3

static_php54 Static/CGI/PHP-5.4

static_php55 Static/CGI/PHP-5.5

static_php56 Static/CGI/PHP-5.6

subversion Subversion

symlink Symbolic link to static/cgi/php app

symlink53 Symbolic link to static/cgi/php53 app

symlink54 Symbolic link to static/cgi/php54 app

symlink55 Symbolic link to static/cgi/php55 app

symlink56 Symbolic link to static/cgi/php56 app

symlink_static_only Symbolic link to static-only app

trac_010 Trac (0.10) - INSECURE

trac_0104 Trac (0.10.4) - DISCONTINUED

trac_011 Trac (0.11)

trac_0114 Trac (0.11.4)

trac_012 Trac (0.12) - Subversion

trac_012_git Trac (0.12) - Git

trac_094 Trac (0.9.4) - INSECURE

trac_100 Trac (1.0) - Subversion

trac_100_git Trac (1.0) - Git

turbogears_088 TurboGears (0.8.8) - INSECURE

turbogears_10 TurboGears (1.0) - INSECURE

turbogears_102 TurboGears (1.0.2) - INSECURE

turbogears_103 TurboGears (1.0.3) - INSECURE

turbogears_1032 TurboGears (1.0.3.2) - INSECURE

turbogears_1042 TurboGears (1.0.4.2) with Python (2.5) - INSECURE

turbogears_1044 TurboGears (1.0.4.4) with Python (2.5) - INSECURE

turbogears_107 TurboGears (1.0.7) with Python (2.5) - INSECURE

turbogears_108 TurboGears (1.0.8) with Python (2.5) - INSECURE

turbogears_10b1 TurboGears (1.0b1) - INSECURE

turbogears_203 TurboGears (2.0.3)/Python (2.5) - INSECURE

turbogears_212 TurboGears (2.1.2)/Python (2.6) - INSECURE

turbogears_233 TurboGears (2.3.3)/Python (2.7) - INSECURE

turbogears_234_27 TurboGears (2.3.4)/Python (2.7)

turbogears_234_34 TurboGears (2.3.4)/Python (3.4)

twisted Twisted (obsolete)

webdav WebDav

webdav_symlink WebDav Symlink

webstat Webalizer

wordpress_342 WordPress (3.4.2) - INSECURE

wordpress_35 WordPress (3.5) - INSECURE

wordpress_351 WordPress (3.5.1) - INSECURE

wordpress_352 WordPress (3.5.2) - INSECURE

wordpress_36 WordPress (3.6) - INSECURE

wordpress_361 WordPress (3.6.1) - INSECURE

wordpress_37 WordPress 3.7 - INSECURE

wordpress_371 WordPress 3.7.1 - INSECURE

wordpress_373 WordPress 3.7.3 - INSECURE

wordpress_380 WordPress 3.8.0 - INSECURE

wordpress_381 WordPress 3.8.1 - INSECURE

wordpress_383 WordPress 3.8.3 - INSECURE

wordpress_39 WordPress 3.9 - INSECURE

wordpress_391 WordPress 3.9.1 - INSECURE

wordpress_392 WordPress 3.9.2 - INSECURE

wordpress_393 WordPress 3.9.3 - INSECURE

wordpress_40 WordPress 4.0 - INSECURE

wordpress_401 WordPress 4.0.1 - INSECURE

wordpress_41 WordPress 4.1 - INSECURE

wordpress_42 WordPress 4.2 - INSECURE

wordpress_421 WordPress 4.2.1

wordpress_older WordPress (older version) - INSECURE

zope2104_30 Zope (2.10.4) - Plone (3.0) - INSECURE

zope2104_301 Zope (2.10.4) - Plone (3.0.1) - INSECURE

zope2104_302 Zope (2.10.4) - Plone (3.0.2) - INSECURE

zope2105_303 Zope (2.10.5) - Plone (3.0.3) - INSECURE

zope2105_304 Zope (2.10.5) - Plone (3.0.4) - INSECURE

zope2105_305 Zope (2.10.5) - Plone (3.0.5) - INSECURE

zope2105_306 Zope (2.10.5) - Plone (3.0.6) - INSECURE

zope2105_312 Zope (2.10.5) - Plone (3.1.2) - INSECURE

zope2105_314 Zope (2.10.5) - Plone (3.1.4) - INSECURE

zope2105_3151 Zope (2.10.5) - Plone (3.1.5.1) - INSECURE

zope2105_316 Zope (2.10.5) - Plone (3.1.6) - INSECURE

zope2105_317 Zope (2.10.5) - Plone (3.1.7) - INSECURE

zope2108_323 Zope (2.10.8) - Plone (3.2.3) - INSECURE

zope2109_331 Zope (2.10.9) - Plone (3.3.1) - INSECURE

zope284 Zope (2.8.4) - Plone (2.1.1) - INSECURE

zope285 Zope (2.8.5) - Plone (2.1.2) - INSECURE

zope293 Zope (2.9.3) - Plone (2.5) - INSECURE

zope294 Zope (2.9.4) - Plone (2.5) - INSECURE

zope295 Zope (2.9.5) - Plone (2.5.1) - INSECURE

zope296 Zope (2.9.6) - Plone (2.5.2-1) - INSECURE

zope297 Zope (2.9.7) - Plone (2.5.2-1) - INSECURE

zope297_253 Zope (2.9.7) - Plone (2.5.3) - INSECURE

zope298_254 Zope (2.9.8) - Plone (2.5.4) - INSECURE

zope298_255 Zope (2.9.8) - Plone (2.5.5) - INSECURE

- *genindex*

INDEX

C

change_db_user_password, 21
change_mailbox_password, 10
change_user_password, 25
create_app, 18
create_cronjob, 19
create_db, 21
create_db_user, 21
create_dns_override, 20
create_domain, 14
create_email, 13
create_mailbox, 10
create_user, 25
create_website, 15

D

delete_app, 18
delete_cronjob, 19
delete_db, 22
delete_db_user, 22
delete_dns_override, 20
delete_domain, 15
delete_email, 13
delete_mailbox, 11
delete_user, 25
delete_website, 16

E

enable_addon, 22

G

grant_db_permissions, 22

L

list_app_types, 19
list_apps, 18
list_bandwidth_usage, 16
list_db_users, 23
list_dbs, 23
list_disk_usage, 9
list_dns_overrides, 20
list_domains, 17
list_emails, 13
list_ips, 26
list_machines, 26
list_mailboxes, 11
list_users, 25
list_websites, 17
login, 9

M

make_user_owner_of_db, 23

P

Packaging, 7
Python Enhancement Proposals
    PEP 257, 8

R

replace_in_file, 24
revoke_db_permissions, 23
run_php_script, 26

S

set_apache_acl, 26
system, 27

U

update_email, 14
update_mailbox, 12
update_website, 17

W

write_file, 24